A Heterogeneous Architecture for Stereoscopic Visualisation

Mark Nelson
mark@itd.dsto.gov.au

Mario Cavaiuolo
mac@itd.dsto.gov.au

Andre Yakovleff
andre@itd.dsto.gov.au

Information Technology Division, Defence Science and Technology Organisation
PO Box 1500, Salisbury 5108, South Australia, Ph: +618 259 5921 Fax +61 8 259 5980

Abstract

This paper introduces a reconfigurable heterogeneous multiprocessor architecture, called Shiva, and describes a configuration of it for generating stereoscopic images in parallel. The main feature of Shiva is its ability to incorporate different types of processors. This allows a computational task to be distributed over the various processors in a way that best uses their combined capabilities. Some results of this system are presented.

1. Introduction

The *Shiva* is a reconfigurable heterogeneous multiprocessor architecture that has been designed to augment general purpose workstations (see Figure 1) by providing specialised processing capabilities [1-3]. In this paper we present a particular configuration that has been designed to generate stereoscopic images in parallel for a terrain visualisation application. This configuration contains two types of processor boards. The first is based on the Intel i860 microprocessor which provides a good balance of floating point, graphics, and integer performance [4]. The second is a more special purpose board called the *Parallel Transformations Board (ParaT)*. The ParaT uses an array of custom VLSI components to perform highly parallel weighted sum operations. In the past these components have been used for building boards to implement artificial neural networks [3,5-7].

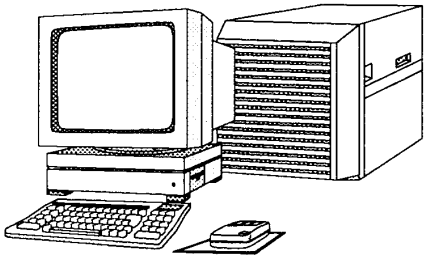

Figure 1: The Shiva augments a general purpose workstation.

The purpose of this work is to evaluate the use of the integrated Shiva+ParaT system for performing stereoscopic terrain visualisation. The visualisation application was chosen to test this architecture because it is a scalable problem and so permits various aspects of the architecture to be demonstrated as the complexity of the application is increased. That is, more realistic images require more computation which can be distributed across more processors. This also allows us to learn in an incremental way how to best use this kind of architecture.

Our approach is to tailor standard graphics algorithms to best utilise the special processing capabilities of the hardware. The i860 microprocessors are used to provide coarse-grained processing for clipping and rendering operations while the ParaT provides fine-grained parallel processing for performing transformations. At this time, the Shiva is fully functional and visualisation programs are being executed on it. The ParaT has been developed but has not been tested.

The remainder of this paper is organised as follows. Section 2 and Section 3 overview the Shiva and ParaT architectures, respectively, and emphasise the features relevant to visualisation. The visualisation process is presented in Section 4 and a configuration of the combined Shiva+ParaT system

0-7803-1363-1/93 $3.00 © 1993 IEEE

is described in Section 5 to meet the processing requirements. Section 6 provides some performance data and discusses the results. Conclusions are presented in Section 7.

2. The Shiva

Shiva is comprised of one *master* board, which handles the interface with the host workstation and the original code partitioning, and a number of *slave* boards, which perform the application. Several modes of communication can be used between the slaves, which do not have to be identical [3].

The i860-based slaves each contain 16 MBytes of memory which can be accessed by any board through a bus. Processors access their local memory through a hotline, hence the memory unit is dual-ported (non-uniform memory access). In addition, slaves can communicate to each other via an asynchronous torus pipeline, implemented in hardware by fast FIFO chips. This is the mode used by the ParaT slave to receive data and transmit results. Transformation coefficients are downloaded onto the ParaT via the bus, therefore the ParaT slave appears as write-only memory (see Figure 2).

The master also contains 16 MBytes of memory and resides on the bus, but has no pipeline connections. Its connection to the workstation, currently a SUN SPARCstation IPC, includes both serial and SBus interfaces. Figure 2 below illustrates the basic architecture and includes i860 based slaves and also non-i860 slaves boards such as the ParaT.

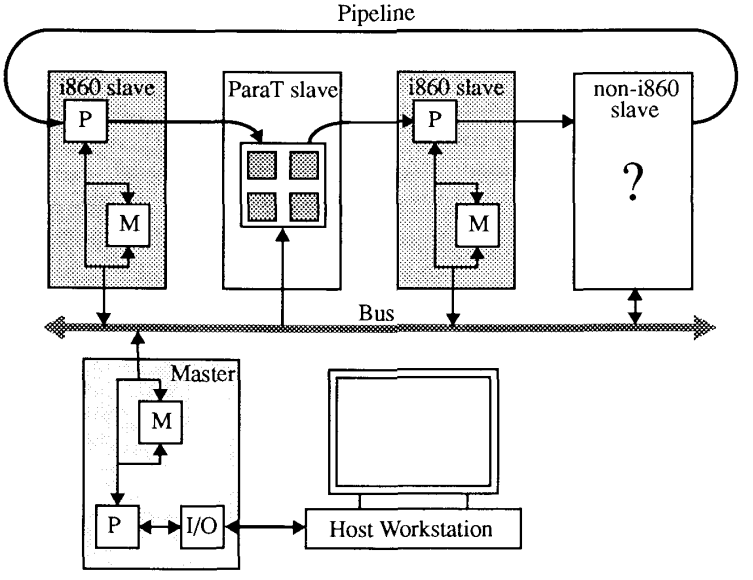

Figure 2: The Shiva architecture. 'P' indicates a i860 microprocessor and 'M' represents a memory module.

All communication paths are 64 bits wide and as they are memory-mapped the system's configuration can be altered dynamically, in the sense that the i860 slaves can work independently like a MIMD machine until such time as a pipeline type of communication, or pseudo-SIMD, is more appropriate.

A fully configured system of eighteen i860 64-bit microprocessors has a theoretical peak performance of more than 1 GFLOPS and 720 RISC MIPS. Sustained performance should be in the region of 400 MFLOPS and 360 RISC MIPS. The peak memory access speed is 80 MBytes/sec. However, this figure is not so critical as the i860 contains instruction and data caches. The maximum pipeline throughput is also 80 MBytes/sec. This is a limitation of the FIFO chips, not of the i860, hence it is expected that a carefully designed program, which also makes use of the processor's internal pipeline, could achieve a sustained performance close to that rate.

The i860 microprocessor has a number of instructions designed to perform operations specific to graphics, such as scan-line rendering, Z-buffering and the 4x4 transforms used for perspective projections [4]. In addition, these instructions make use of the 64-bit wide data path to perform operations on several pixels simultaneously, depending on the size of their representation.

3. The ParaT

The ParaT comprises eight custom neural network units called *Neural Accelerator Chips (NACs)* which perform weighted 10-bit multiply/accumulate operations, at a total of 4 GOPS. Each NAC is a linear systolic array architecture comprising sixteen neural processing elements, called *Neural Accelerator Processors (NAPs).*

Figure 3(a) shows that a NAP consists of a multiplicand (data) buffer, sixteen weight registers, a multiplier and an accumulator.

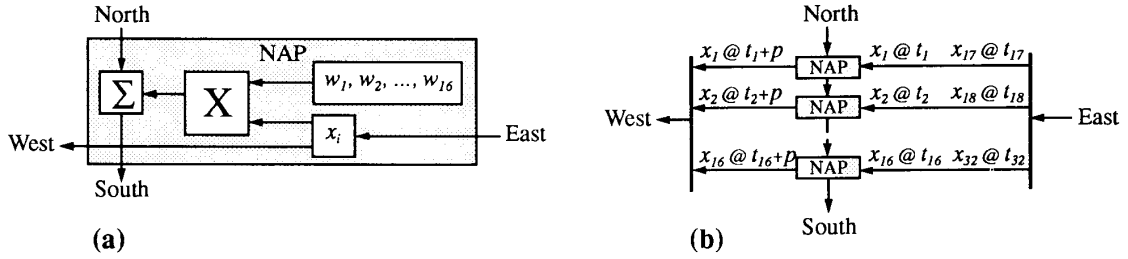

Figure 3: Data flow diagrams of the NAP and the NAC.

If at cycle i, x_i is the data, w_i is the weight (which are cycled through every sixteen clock cycles), $North_i$ is the result from another NAP at cycle i-1, then the result at cycle i+1 is

$South_{i+1} = North_i + x_i w_i$.

The NAPs are arranged linearly within the NAC such that the data required for processing flows from east to west while the accumulated resultant data flows from north to south, as illustrated in Figure 3(b). This feature allows chips to be cascaded along both the east-west and north-south axes.

On each cycle i (denoted by t_i) the n^{th} NAP multiplies its multiplicand buffer value (distinct values denoted by x_j) by its k^{th} weight $w_{n,k}$. This result is then added to the resultant data which comes from the processor immediately before it. The accumulated sum is then fed to the next processor in time for the next clock cycle. Thus, the resultant output from the NAC is

$South_{17} = North_1 + x_1 w_{1,1} + x_2 w_{2,2} + \ldots + x_{16} w_{16,16}$

$South_{18} = North_2 + x_1 w_{1,2} + x_2 w_{2,3} + \ldots + x_{16} w_{16,1}$

Figure 3(b) illustrates this data flow through the NAC and also shows the west output as a delayed east input, where the phase delay is controlled externally via the phase input p.

Each NAC calculates the pipe-lined sum of products at a rate of 500 MOPS at an operating frequency of 16MHz. On the output of each NAC exists switchable hardlimiting circuitry, which serves to incorporate the non-linearity inherent in the output of biological neurons.

The ParaT design exploits the systolic nature of the NACs to increase the parallelism of the computation. The ParaT (shown in Figure 4) can perform multipler 4x4 convolutions at frame rates upon the incoming data. The inputs to the ParaT are from the 64 bit wide Shiva pipeline from which four 10 bit words at 10MHz are fed into the East inputs of the NACs. Each module of four NACs is an array that performs a single transform on the input. The transformations performed by the NACs have the form

$x' = x.w_1 + y.w_2 + z.w_3 + w_4$

$y' = x.w_5 + y.w_6 + z.w_7 + w_8$

$z' = x.w_9 + y.w_{10} + z.w_{11} + w_{12}$

where (x,y,z) are the input point coordinates, (x',y',z') are the transformed coordinate points and the transformation coefficients, or weight values, are (w_1,w_2,...,w_{12}).

At the output the data is fed off the ParaT and back onto the Shiva pipeline. The weight values for each of the NACs are loaded onto the board via the Shiva Bus.

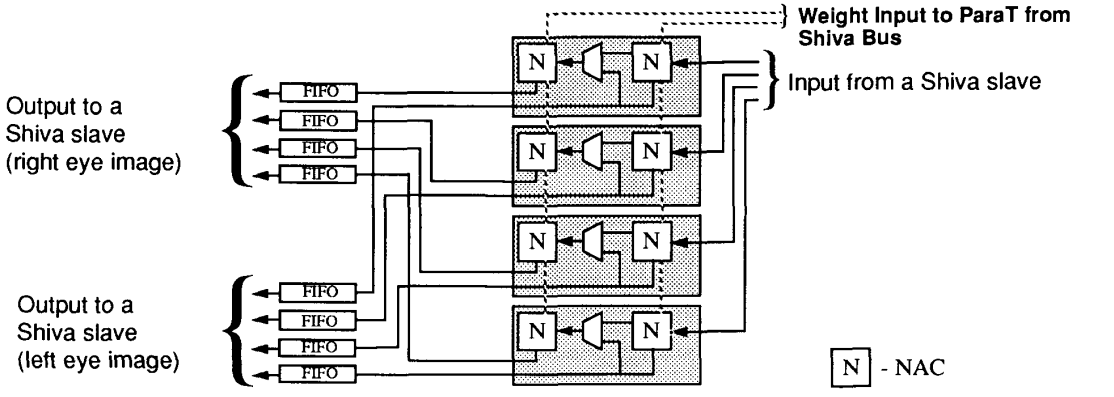

Figure 4: The ParaT architecture.

Hardware Configurations

Figure 5 shows the two possible configurations of the ParaT in performing different transformation tasks. The schematic of Figure 5a shows how the north-south data flow can be used to serially apply two transforms T_i to an input data stream. Alternatively, the configuration shown in Figure 5b shows how the east-west data flow can be used to apply two different transformations in parallel to the same data stream. This demonstrates the flexibility of the NAC's connectivity.

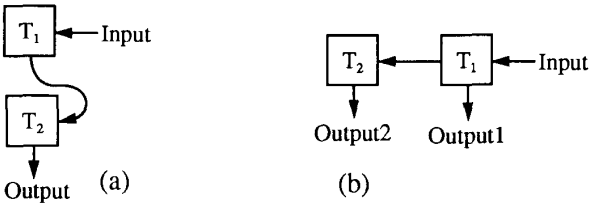

Figure 5: ParaT configurations.

4. Stereoscopic Terrain Visualisation

The terrain is represented by a digital terrain model (DTM) which comprises a digital elevation model (DEM) and a digital attribute model (DAM). The DEM is a 2D grid of height values and the DAM is a 2D grid of visual attribute values (e.g. colour and texture). Thus, the height and visual appearance of each point in a plane can be obtained by accessing the corresponding elements of the DEM and DAM, respectively.

The task is to compute stereoscopic views of the DTM by generating perspective images for the left and right eyes of an observer so that depth information can be perceived when the images are viewed together. The challenge is to implement an image generation scheme that best utilises the combined capabilities of the i860 slaves and the ParaT. The ParaT will be used to perform coordinate frame transformations for both left and right images in parallel while the i860 slaves will be used to perform the clipping and rendering operations.

The ParaT and i860 slaves have pipeline interfaces and so are well suited to image generation which is innately a parallel pipeline process. Figure 6 illustrates the image generation pipeline being employed. It is a simple coarse-grained non-linear pipeline incorporating fine-grained parallelism at

the tranformation stage. Section 5 presents a configuration of the hardware for implementing this scheme.

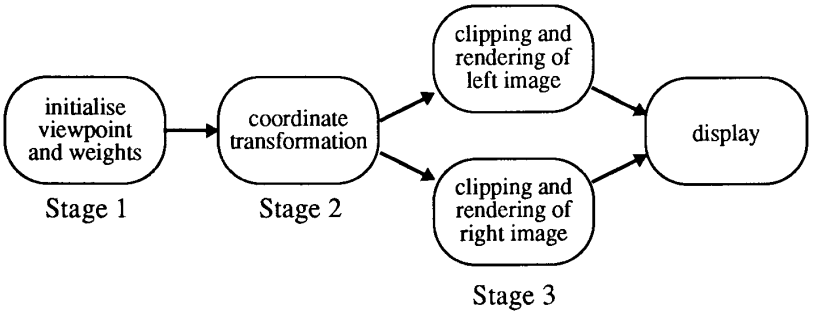

Figure 6: The image generation pipeline.

In stage 1 the position and orientation of the viewpoint are computed relative to the coordinate frame of the DTM. Then the transformation weights are initialised for stage 2. These weights also incorporate the zoom factor and focal length to reduce the computation in subsequent stages.

In stage 2 the weights are applied to a stream of DEM points to produce an output stream of transformed points whose coordinates are relative to the viewpoint.

In stage 3 the stream of transformed points are then clipped using the Liang and Barsky parametric line-clipping algorithm [8] and the line segments drawn using the Bresenham line-drawing algorithm [9]. The resulting wireframe representation of the DTM is then displayed.

5. The Shiva+ParaT Configuration for Stereoscopic Visualisation

The scheme described in the previous section is easily implemented on the Shiva by pipelining several slave processors as shown in Figure 7. The following briefly explains the operation of each processor board in this configuration.

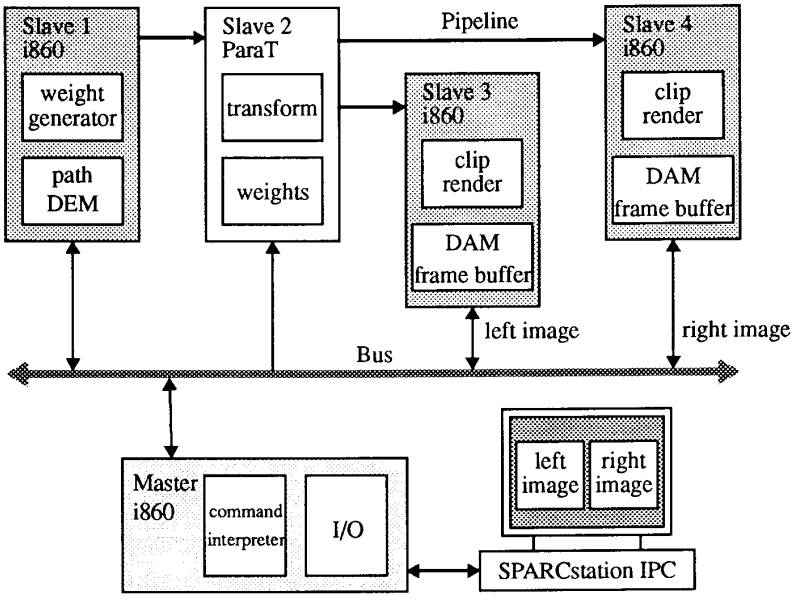

Figure 7: The Shiva+ParaT configuration for stereoscopic visualisation.

Firstly, the master is responsible for interpreting commands from the host workstation to set visualisation parameters (e.g. zoom and focal length) and to initialise the DAM and DEM, etc. The master will also copy the completed left and right images from the slave 3 and 4 frame buffers, respectively. These images are then sent over the SBus to be displayed on the workstation.

Slave 1 is responsible for determining the next viewpoint according to a specified trajectory and computes the left and right transformation weights. These weights are loaded onto the ParaT via the bus and then slave 1 will pipe the DEM data into the ParaT.

Slave 2 is the ParaT and applies the two sets of weights to the input data stream and produces two output streams, one for the left image and one for the right image. This process utilises high parallelism to perform the 4x4 transformations and does so to the two images simultaneously.

Slave 3 and slave 4 clip the transformed input and then combines it with the corresponding DAM data to render the left and right images, respectively. These images reside in frame buffers local to those processors.

Various techniques are being experimented with to increase the throughput. For example, slave 3 and 4 use two frame buffers each so that a new image can be created while the master is sending the previous one to be displayed. Also different caching schemes are being used which is not a trivial problem in a multiprocessor, shared memory architecture.

6. Results

This implementation of the visualisation application has occurred in a number of stages. Each successive stage is intended to improve the quality of the visualisation process (ie. realism of images and speed of generation) and this consequently increases the sophistication of the hardware configuration and the application software.

In stage 1 only the Master board was used to generate point images. This was done at a rate of 3 256X256 images per second.

In stage 2 an i860 based slave board was introduced. The slave generated the images in a double frame buffer so that as one image was generated, the previous one was being read by the Master and sent to the host workstation. A similar rate to stage 1 was achieved.

In stage 3 the computational load was increased by generating wireframe images (such as is shown in figure 8) using the master and one slave. A rate of 2 seconds per image was attained.

In stage 4 a second slave was added to facilitate a pipeline. Slave 1 transformed edges and points and piped them to slave 2 which clipped and rendered these edges. The Master read the images from the double frame buffer in slave 2 and sent them to be displayed on the host. A hardware fault causing noisy images was detected and has been corrected. A throughput of one image every 0.4 seconds was achieved for this stage.

Example Data and Simulation Results

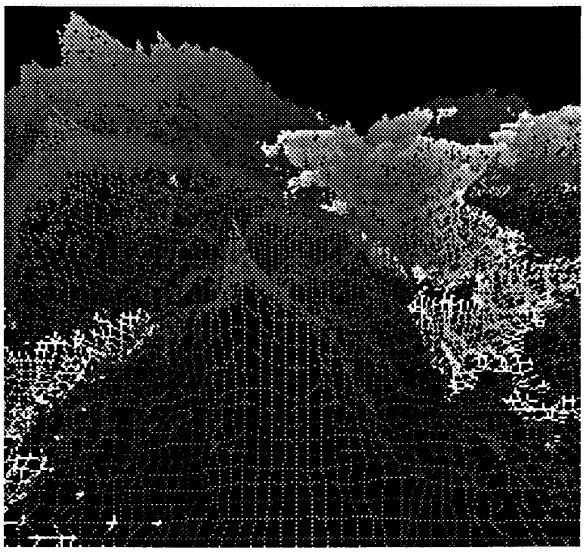

Figure 8: An example wireframe image generated by Shiva.

In stage 5 the ParaT is to be introduced and two more i860 slaves added to generate stereoscopic images as described in section 5. The ParaT board has been manufactured and functionally tested, and once introduced will perform the point transformation part of the visualisation processing at a far greater rate than was achievable by the i860 slaves. This stage has yet to be completed and tested at the time of writing this paper.

Further stages will generate more realistic images to those shown in Figure 8, and be able to reconfigure the hardware, and redistribute the software as required. As is evident by our results the heterogeneous architecture can be scaled up further with more ParaT, i860 slave and other specialist processor boards until the required quality of image and speed of processing is reached.

7. Conclusion

The Shiva has been described as a heterogeneous multiprocessor architecture that allows highly specialised processors like the ParaT to be integrated with general purpose processors such as the Intel i860. A scheme for a stereoscopic visualisation pipeline was presented that best utilises the combined processing capabilities of the Shiva and ParaT. The novel feature of the Shiva+ParaT system is that it incorporates fine-grained high parallelism processing within a coarse-grained nonlinear pipeline. Though no significant visualisation results have been provided, the main achievement here is the demonstration of a unique heterogeneous multiprocessor architecture that can be reconfigured to provide increased processing power in order to improve the realism in the images generated. Ultimately this will lead to realistic stereoscopic visualisation.

8. References

[1] M.S. Anderson and P.C. Drewer, *Design Overview of the Shiva*, IEEE Region X Conference, Hong Kong, 1990.

[2] A.J.S. Yakovleff, J.D. Yesberg, D.A. Krnak, M.S. Anderson and P.C. Drewer, *An Expandable Supercomputer Architecture with Dynamic Reconfigurability Properties,* Fourth Australian Supercomputer Conference, pp 175-183, Gold Coast, Australia, December 1991.

[3] M.S. Anderson, J.D. Yesberg, A.J.S. Yakovleff, D.A. Krnak, P.C. Drewer, C.R. Watson and M. Cavaiuolo, *A Heterogeneous Parallel Accelerator for Image Analysis and Radar Signal Processing,* HICSS-25, Hawaii International Conference on System Sciences, pp 129-138, Feb. 1992.

[4] N. Margulis, *i860 Microprocessor Architecture.* Intel Corp. and Osborne/McGraw-Hill, 1990.

[5] C.R. Watson and M. Cavaiuolo, *A Neural Accelerator,* Proceedings of the 9th Australian Microelectronics Conference, pp 115-120, Adelaide, Australia, July 1989.

[6] M. Cavaiuolo, A.J.S. Yakovleff and C.R. Watson, *A Systolic Neural Network Image Processing Architecture,* Proceedings of CompEuro'92, The Hague, The Netherlands, May 1992.

[7] M. Cavaiuolo, A.J.S. Yakovleff and C.R. Watson, *NAC: A Building Block for Neural Architectures,* ICANN'92, International Conference on Artifical Neural Networks, Brighton, England, Sept. 1992.

[8] Y.D. Liang and B.A. Barsky, "A New Concept and Method for Line Clipping", *ACM Transactions on Graphics*, vol 3, no 1, pp 1-22, January 1984.

[9] J.E. Bresenham, "Algorithm for Computer Control of Digital Plotter", *IBM Systems Journal*, vol 4, no 1, pp 25-30, 1965.